Release 1.0

88-3 15 March 1988

SHARING IDEAS

Add one more category to the "wares": mysteryware. At the PC Forum Jim Manzi announced Notes, Spencer Leyton announced GrandView, and Mitch Kapor coined the term mysteryware to describe them and his own new product Agenda. While we wax rhapsodic over the joys of groupware, text-based applications and OS/2 server applications, most vendors (let alone users) don't even know what they do.

That's why most mysteryware is also vaporware: The danger is not that people will steal these ideas; to the contrary, they must be lured to accept them. Their vendors -- Lotus and Persoft (IZE) among others -- generally announce them early, in hopes that users will be ready by the time they ship. But Symantec's GrandView, a few new ideas dressed in more familiar clothing, will ship this month to a market ready to receive it.

In talking about groupware, we're still a little like sundwellers talking about snow who haven't yet learned the Inuit (Eskimo) words for all its varieties. The fundamental components of groupware -- its hydrogen and oxygen, so to speak -- are data transfer and data storage. All snow or groupware is built from these; beyond that, structure and impurities give snow its character, just as architecture and ancillary functions give groupware *its* character.

To take the two extremes of groupware: E-mail is mostly pure data transfer, with a data structure that consists of a list of files. It's all action and no content; messages are delivered intact, their content unexamined. E-mail's complement, a multi-user database management system, is all content and no action; users or applications must transfer the data.

The distinguishing components of groupware are different data structures and data-manipulation routines. In Lotus Notes, for example, the basic data structure is the note, an arbitrarily long field of text and graphics that contains name-value pairs (such as "To: <NAME>") that work like fields in a traditional rela- ⟹

CONGRATULATIONS GREG & JUDY!!

INSIDE

SHARING IDEAS	1
Groupware and textware.	
THE IMPACT OF OS/2	5
Indiscrete applications.	
DATABATTLES	7
Report from the front.	
NITTY GRITTY EXPERTS	9
The intelligent database.	
NAMED PIPES AND ALL THAT.	11
Across the wires.	
INTERFACE ISSUES	15
Don't manipulate; delegate!	
Tandy's DeskMate.	
CASE IN CONTEXT	18
What users want.	
RELEASE 0.5	19
PC Contents™, CD ROM,	
Softeach, Toon titles.	
RESOURCES/CALENDAR	20/21

tional database. Notes (records) can be selected or matched according to criteria in the fields, for display in rich formats most meaningful to users. For example, memos could be sorted by day, by subject, or by any relevant attribute. Data in each category could then be sorted and selected further. The resulting feel is that of an outline, with a hierarchy displayed not so much to the requisite *level* of detail, but with the requisite *layers* of detail. For example, the top line of the outline can be dates (as in a correspondence file), with the text searchable by other fields, such as topics, etc. Multiple users can add their own comments and data. However, data within a given note always "belongs" to that note; the system does not do joins (or unions of databases), just selections (or subsets). It can select items, but it cannot easily join or restructure data (change A1, A2, A3; B1, B2, B3; C1, C2, C3; to A1, B1, C1; A2, B2, C2; A3, B3, C3).

How does all this happen? In fact, Notes is not an application any more than, say, dBASE is. It's a tool for building multi-user information management systems, with much of the style and content determined by the builder-user. The end-user simply executes a number of pre-defined queries and uses the system's formats as a guide in entering data.

For example, consider a hypothetical personnel department. One large section of its database concerns job candidates; another contains open positions. Data about the candidates (one note for each) consists of resumes and other material submitted by the candidate, comments from interviewers, test results, reference checks and the like, as well as more structured data such as previous employment and qualifications. Ultimately, each candidate ends up with a substantial file (note). At specified intervals, closed notes are sent to an archive or selectively destroyed. The open-positions file is a little more structured, but also contains room for free-form text about the job and its peculiar challenges.

The whole database is open to the personnel department; personnel people use it to record their comments about candidates and to filter in the information they need. Managers in other departments have access to it to post job listings and to notify personnel of employees ready for promotion or wanting a transfer. The system attempts to match candidates and openings according to criteria set by the system designer in conjunction with the personnel managers (but this isn't quite as automatic a process as it would be with a relational database). Obviously, all the details of the system are up to the designer: What data is collected? How are candidates and jobs matched? Who has authority to use what parts of the system? How are discipline problems, salaries, controversies recorded and safeguarded? What happens to candidates' files when they're hired?

The purpose of Notes is to make the design of such a system easy and to manage the necessary underpinnings: file access, display of data in a meaningful way (with heavy use of outlining features such as showing and hiding detail), notification of the appropriate users when new data is added, text search and matching, standard database functions, and authentication (not just, Who may read this? but, Who actually wrote it?).

Notes designer Ray Ozzie, who founded Iris Associates to develop Notes for Lotus after working on Symphony there, built the product in order to "foster communications a heck of a lot better than mail. When you get flooded with mail, you tend to ignore it all. With Notes, with information sorted by

topics and with notifications, you can look at stuff when you're interested. You tend to use mail only for personal, hot stuff, and the rest of the communication is organized."

Discrete applications

By contrast, Agenda is a *personal* tool, and treats data as a far more granular substance. This granularity provides for extreme flexibility in Agenda, but it would be hard to achieve in a networked system such as Notes. While Notes generally strips away extraneous data to select only the portions that are of interest from each note, Agenda joins pieces of data as specified by common values and other criteria to create views. Views in Agenda are generally agglomerations of data lumped together through common links to other pieces of data.

The obvious question is, Why are Notes and Agenda two separate products? The short, lame answer is that they were developed by two separate teams, Iris Associates and Ed Belove-Jerry Kaplan-Mitch Kapor, both consisting of former Lotus employees (except Belove, who's still there) who left to develop their products independently but still chose to publish through Lotus.

But the longer answer is that they *should* be separate products under OS/2 ...as long as Lotus can develop some common data interchange formats so that they are in fact simply two different approaches to manipulating the same underlying information. Agenda items, for example, could be assembled into Notes. Then Notes could offer the multi-user, rich-interface, message-sending components of a broad, modular system, while Agenda would supply the automatic data structuring and personal-organization facilities, based on its rules language. All this will be quite possible under OS/2 -- with a bit of work, to be sure.

ViewStar: aiming high

ViewStar (Berkeley, CA) is yet another company in the mysteryware business. It will be showing its product Pro Series at the AIIM image-management show next month, although in its debut Pro Series will look more like an image-management application than the groupware-building tool it is underneath. That is, ViewStar will be showing what you can build with Pro Series rather than the system itself. While Pro Series includes text editing and a powerful graphics program, it addresses management of documents and images *after* their creation, when they must be passed from hand to hand, revised, annotated, amended, approved all within deadlines. It deals not with What-if? but with What-next?: Who gets the items? Who revises them? Who makes which changes? Where do they go when they're completed? And when?

The system includes a rich networked environment including a variety of document- and image-manipulation tools plus two distinguishing capabilities: an object-oriented file system, VS*Star, and a macro language, VS*Script. VS*Star enables the Pro Series to manage all kinds of data and links among data easily, and VS*Script enables builder-users to construct scripts, set up multi-person procedures and otherwise customize the system. Much as in HyperCard, users can write their own routines, or scripts, which can be activated by selecting an appropriate object on the screen. The objects Pro Series works with have generally been created elsewhere -- by a CAD/CAM system, say, or a word-processor -- and are now ready for mark-up, editing, and

further paper-shuffling. Although users can interact with the data within each file/object -- editing a document, say, or modifying a drawing -- the system itself deals only with the objects as a whole, and can identify them only by externally recorded attributes (creator, title, etc.). Long run, it will be able to examine their contents (as can, say, Staffware; see **Release 1.0**, 30 December 1987) in deciding how to handle and route them.

Working at Impell, an engineering management consulting firm, colleagues Kamran Kheirolomoom and Russell Rowe "saw customers struggling with their paperwork." They left in 1986 to found ViewStar and to build an internal development environment for building CAD image libraries. Thus Pro Series has been built on the job with input from the engineering groups of several large utilities and manufacturers, and includes many ad hoc tools and facilities, not all of them fully documented or positioned. The system is much richer than can easily be described -- or marketed. The applications demonstrated at the AIIM show will handle such tasks as engineering drawing management, and processing of insurance applications and bank loans.

Built in LISP underneath, Pro Series has its own English-like user language. It runs on networked PCs and fits easily into existing environments. An installation starts at $15,000 to $20,000, and costs from $5000 to $8000 per active user as monitored by the system (with an average of four to five active users per workstation). The company has venture funding from Hambrecht & Quist and from several Impell founders, among others.

GrandView -- a familiar view

Symantec's GrandView, also announced at the Forum, provides an easier transition to novelty than most mysteryware. Essentially an extension of the outlining model (with word-processing and publishing facilities included), it lacks the powerful underlying data structure of, say, Agenda, which is based upon a relational database. GrandView is the answer to people who say, "Gee, I wish I could reorganize this outline automatically, instead of having to move things one at a time, by hand. [This is the burden of direct manipulation, as discussed on page 15.] For example, if I could mark some of these items and have it select them... Or just list things by date instead of by topic..." These people's prayers have been heard. GrandView will let them mark items -- easily, with string-matching or the press of user-defined key combinations -- and then sort their data by the markings. What it will *not* do -- and Agenda does -- is mark or sort text by more complex criteria, such as any item about Juan but not about Alice, or a list of hotels by city within the state of California.

GrandView supports the principle of least astonishment: At all times the user knows exactly what the program is doing. By contrast, mysteryware users aren't always sure what their software is doing, and they're not yet ready to take it on faith. Of course, VisiCalc once looked mysterious too, but now half the people you meet on an airplane are prepared to discourse at length on the relative merits of 1-2-3, Excel and Quattro. The mysteryware snow will one day crystallize clearly into a variety of types and models.

THE IMPACT OF OS/2

While everyone acknowledges the importance of OS/2 and many people talk about Presentation Manager as if it alone were OS/2, they are missing much of the point -- just like people who think all they have to change to live overseas is the words they use. In fact, OS/2's effects will ultimately extend beyond the boundaries of single applications and alter not just systems architecture but patterns of usage and even distribution of software. As we noted above, it is not necessary that Agenda and Notes, for example, be combined into a single product, but only that they should share data formats and function semantics, so that the facilities of one can be used by the other, and so that they can work cooperatively on the same data. Such integration is fairly tough to arrange nowadays, and is most easily achieved by a single vendor with control over both products. Now, however, with the help of design rules and tools such as LEAF and the Dbase language, a number of add-in vendors are building cooperative applications for 1-2-3 and dBASE.

On a broader scale, with OS/2 and common data structures and a common language, be it Extended BASIC, REXX, or some other (see **Release 1.0**, 30 December), cooperative applications will be widespread and users will have great flexibility in assembling modular tools to do their work. "Applications" for less sophisticated users may consist of front-ends that do such assembly for them, as "applications" call on modular facilities in a dynamic linked library, and pass data from one process to another, much as applications now call on DOS for file management. A UNIX spell-checker, for example, consists of a combination of UNIX routines, first sorting all the words in a document alphabetically, comparing them to a dictionary, then displaying the mismatches with pointers back to where they occurred. (It's not exactly a model of friendliness or interactivity, but it illustrates our point.)

In the future, developers may assume or require the presence of modules beyond the operating system itself -- and not just Extended Edition or printer drivers, either, but, say, standard wp functions that could be called to work within the fields of a forms processor (like the Mac's desk accessories, but more integratable). Applications will consist of selections and sequences of other people's code, managed and catalogued for easy identification and linkage by an object-oriented file system. As we move towards reusable code, the *selection* of chunks to reuse, rather than their *creation*, will become an important function of design. Cooperation among vendors and the adoption of standards must prevail.

OS/2 1.X -- the installable file system

As we've noted earlier, OS/2 will eventually be enriched/extended with a cross-application macro language, which will solve (to the extent that it's adopted) the problem of "dynamic command exchange." In addition, Microsoft/IBM will add an installable, object-oriented file system, to replace the current flat files with the eight-character names and three-character extensions. An installable file system is also deinstallable; that is, an installable file system operates independently of the OS and can be modified independently. Moreover, multiple file systems can be installed, allowing a variety of different data structures to coexist. (One of the first would-be examples is Novell's Nondedicated Netware, which will operate as a task under OS/2 and supply its own file system. However, it doesn't use OS/2's

file-installation model, which doesn't yet exist, but simply bypasses the file system that exists.)

Among other benefits, installable file systems will be able to look like the current rash of directory managers that characterize files by author, topic and other attributes. These directory managers are in fact a little database with pointers to the files in question. Their information is captured and saved in a separate application.

With an object-oriented file system, each file can maintain its own data about itself, and can assemble the appropriate executable code modules for an application. Applications will no longer look like a bunch of .EXE files to the file system or user, but will instead be stored (logically) as procedures called by any of the data objects they act upon. (How they are stored physically will be managed invisibly by the file system, as it is now.)

Sales support

Rather than "open" your word-processor, for example, you may select a document from a folder or tear a fresh sheet off a "writing pad" object, thereby automatically invoking document-creation functions. Perhaps one instance of the writing pad will be specialized for writing sales reports and will automatically load the latest week's sales figures (complete with chart). All you have to do is look at the figures and explain why popcorn sales were down, before the system sends the memo out to your current boss. Such virtual applications, consisting of the cooperative action of application modules, will be easily managed by the objects themselves, which will "know" where to find the appropriate modules, whether beforehand or interactively. Pick "layout" from the menu generated automatically by the document object, and you'll call the layout tools.

As models of what might be, consider first the Lisa, which offered a complete suite of applications attached to the appropriate objects but failed for other reasons, and Hewlett-Packard's NewWave. NewWave is a good model for the future, but it suffers from being single-user (though networkable) and DOS-based, and from not coming from an acknowledged vendor of "standards." These system foundations are the sort of facilities that are almost useless if they aren't standard, since they offer benefits mostly to the extent that they foster cooperation among a large number of applications and consistency for a large number of users.

Of course, if this were all so easy, we'd have it today. Microsoft and IBM have to solve a variety of problems to make it work. The file system will have to register and manage a class for each set of objects, and manage links between those objects and multiple application modules. Objects gain their value by being customizable, but the system must be able to distinguish between modifiable objects and the sacrosanct originals (classes) from which they inherit their structure and behavior (code). Ultimately, OS/2 and its networking environment(s) will have to know the location of objects and code throughout a network and even outside it, in order to provide transparent application and communication services to their users.

DATABATTLES: REPORT FROM THE FRONT

The Ashton-Tate/Microsoft/Sybase alliance is not quite a clean sweep: Aside from IBM and Oracle, there are still a few other contestants such as Lotus, Borland/Ansa, and the dBASE pilot fish.

The dBASE pilot fish (**Nantucket, WordTech, WallSoft, Foxbase, et al.**) generally point out that dBASE IV now offers the facilities that they pioneered, and that they too will support SQL in due course. In due course, we expect to see the nimbler of these companies move towards supporting SQL Server with application-building tools, just as they once did for dBASE itself. However, that doesn't mean that they'll abandon dBASE; for all its wonders, SQL is only a data language. You still need the Dbase language (or C, or whatever) to write the applications that use the data.

To be sure, the competitive balance will shift back in Ashton-Tate's favor now, but we don't see much impact either way. The pilot fish, however, will find tougher and more widespread competition in the SQL server tool market than they ever got from Ashton-Tate or each other. Sybase, unlike Ashton-Tate, will be there competing with its third-party "supporters" from the start.

Lotus has offered some more details on its new database management system, Lotus/DBMS, with some help from **Gupta Technologies**, which built and sells SQLBase, a PC server engine, and has reshaped a version of it to conform to Lotus's (customers') needs. Appropriately, Lotus's interest in the database market is primarily derivative: It wants to sell its tools, especially 1-2-3/G (which can also work standalone), and other tools that will probably remind the world of Metaphor's superb suite of graphical data-modeling tools in their intuitiveness and power.

Thus, we don't expect to see Lotus spending a lot to convince the world that Lotus/DBMS is the world's best general-purpose engine. If its users need a database server, Lotus will be happy to sell them Gupta's, which can run both locally, along with the tools on a PC, or as a networked server application. (Gupta stresses that its SQL dbms is unique in being developed specifically to run on a PC, rather than being squished down from a mini version or reworked from a non-SQL PC dbms.) Lotus/DBMS is a good vanilla server enhanced with extensions to make it uniquely suitable for spreadsheet and decision-support applications. These include read consistency, so that all the data in a spreadsheet are as of the same time rather than selectively updated (as in closing the quarter: you'd rather have all the data out of date consistently than some last-minute and some old); scrollable fetches (so that you can scroll both up and down, a feature still lacking in some systems); and other more common niceties such as editable queries and read-only mode (to avoid locking up files).

Aside from these, the major feature of Lotus/DBMS (and Gupta's SQLBase) is its ability to act as an interface to other databases, notably DB2 and OS/2 EE, the SQL Server, the Oracle server, and eventually others. Lotus will shortly announce a widespread data access architecture to enhance connectivity, both among its own number and data products, and to other sources of data. Like SAA, it won't be so much a product as a set of specs that others will be encouraged to adopt. They may adopt and support it just in order to get at the 1-2-3 installed base, and thereby enhance the appeal of Lotus/DBMS and the Lotus tools as well. Clever Lotus!

The company with the most to lose -- because it *has* the most right now -- is **Oracle**. Oracle has achieved stunning success in the minicomputer market, with its so-called "one size fits all" strategy. In fact, its strategy has been "one fits all sizes." Oracle has helped to establish the IBM standard across a wide variety of hardware platforms. But the marketplace is shifting. In the new world, the issue is not just using mini and mainframe data for decision support (Oracle's current and Lotus's soon-to-be strength), but using 386 platforms for industrial-strength production/transaction-oriented tasks, and cooperating with, but not necessarily executing database functions on, PCs. (Oracle has some other items up its sleeve, however.) The company would have liked to join forces with Lotus, but Lotus, unlike Microsoft, may feel that it can't get away with competing with IBM and would rather remain noncommittal all around. There's more to a relationship than something you can announce, and Lotus doesn't want to jeopardize its connections with IBM.

Borland/Ansa is taking something of the Lotus position, saying in effect, "We'll support everything," but without retreating to surfaces-only. Paradox will be able to communicate intimately with SQL Server and OS/2 Extended Edition's Database Manager, but (for now) it will retain its own engine. Unlike Sybase with its SQL Server, Paradox does not strongly follow the client/server model. That is, Ansa will not seriously propose the Paradox engine as something for which other companies can write front-ends.

DB/Access

All these solutions, however, ignore a fundamental reality of the marketplace: Most of the data isn't in SQL-based databases in the first place. The familiar diagram across (numbers approximate) shows the importance of talking to foreign databases. Although the workstation end of the market is growing much faster than the mainframe business, most of the data is still sitting in mainframe disk farms, not in DB2 but in IMS or IDMS or VSAM even flat files. That's where a product such as **DB/Access**'s VIEW/PC comes in. Taking the pragmatic point of view that the database market is already amply served, DB/Access is offering a high-level connectivity tool that enables users to extract data from old-style databases as well as from the more modern relational ones everyone seems to handle mowadays. The process isn't easy, because it's not a question of a one-for-one translation of a couple of commands: The fundamental data structure is different. DB/Access has just signed a contract with Businessland for distribution of VIEW/PC.

	Data	MIPS
Mainframes	80	20
PCs	20	80

DB/Access basically says, "You do the database management. We'll help you get hold of the data." Financiers and other vendors are certainly looking at the company with new respect since the recent acquisition by Apple of **Network Innovations**, which provides the same sort of links between pcs and Macs and most SQL databases. Network Innovations has also just introduced CL/1 (Connectivity Language 1), an APPC/LU 6.2-like mechanism (see page 14) for manipulating remote databases that now works between Macs and VAXes and will be extended to other systems soon.

NITTY-GRITTY EXPERT: AICORP'S INTELLIGENT DATABASE

We generally think of SQL as a database language, but it is both more and less than that. It does not build applications, per se, but it can manage data not just for traditional database applications but also for project managers, mail systems, even expert systems. In project managers and the like, SQL provides a way to get at and manipulate the data; in e-mail, you can build structured messages (a.k.a. forms) and select and manipulate messages according to data in fields of the forms (more on this next month). In expert systems, you can use it to manipulate not just naked data but richer objects that carry a lot of baggage with them -- all unknown to the SQL syntax that manipulates them.

As it happens, this is not just a neat idea (such as the other suggestions above), but an actual product from AICorp, shipping this summer and now in beta at seven sites. AICorp's KBMS, for Knowledge Base Management System, represents in a neat way the confluence of two technology threads: Founder Larry Harris is a recognized expert in natural language and in artificial intelligence in general. Bob Goldman, by contrast, was an expert in database technology at Cullinet, a high-flying company whose very success shielded it from knowledge of AI and other technologies apart from its own. When these blinders and other mistakes led to the company's faltering, Goldman shouldered more than his share of the blame. During six months of job-hunting in 1986, Goldman gained a broader perspective, discovered AI, and joined the company then called Artificial Intelligence Corp. as ceo.

Goldman already knew Harris from a deal whereby Cullinet resold AICorp's Intellect natural-language front-end for databases. The two began an ambitious project to build a knowledge base management system, which would have a similar relationship to a rule-based expert system as a database management system has to an application. That is, KBMS provides not only the logic, but also a way to store rich data, describe its relationships, and maintain it separately so that it is usable by a variety of discrete expert systems or applications. The KBMS also includes control structures and compiled rules that are the inference engine equivalent of indexes and stored procedures in enhancing performance and optimizing use of resources.

SQL provides the query language to transfer data across the boundaries -- from the database and the knowledge base to the user interface and the applications. In essence, the architecture doesn't look all that different to a user or an application from a traditional database-based environment, with the exception that some of the data is "alive," stored as objects in the knowledge base instead of as data in the database. KBMS accomplishes this with an object-oriented model: That is, the data has rules and structure attached to it, which can be executed by the knowledge base. (Some of these same ideas are found in the Sybase SQL Server, with its stored procedures and triggers, but it lacks the hierarchical structure and inheritance features, as well as the richness of the English-like rules language, of KBMS.)

Contrast and compare

Like its competition from Aion and Cullinet, KBMS works within all the standard IBM mainframe environments, but it lacks (so far) a PC-based front-end. The major difference between KBMS and competitors Aion's (ADS) or Cullinet's (Application Expert) approach is its strong object orientation and AI fla-

vor. Although all ultimately perform much the same functions, KBMS attaches procedures (code) to the data, whereas Aion and Cullinet call the appropriate expert system code as part of an application. That is, if a KBMS application asked for a price via an SQL query, KBMS's data dictionary would know whether the answer was data from the database or a value determined by the knowledge base, and send the query to the appropriate module for execution; in the Aion or Cullinet version, a statement in the application would directly call the price-determination routines.

The expert system logic sits in the knowledge base, and is activated whenever an application or a user asks (via SQL) for something within or determined by the knowledge base. For example, "select employees where department = kidgloves" might get a list from the database, whereas "select employees where performance = outstanding" might fire off a set of complex definitions and comparisons in the knowledge base, which would use data from the database as necessary. In the sample grocery-bagging application we saw, for example, all the items to be bagged (with dimensions and weights) were in the database, while the actual items ordered and the bags to fill were knowledge base objects with specs inherited from the database and behavior from the knowledge base, subject to such rules as "Pack heavy things first" and "Sell the user Pepsi if he buys potato chips." The appropriate SQL statements might be "Insert" a list of the items to be purchased, followed by a query as simple as "Select name and bag-no from items."

There is an extra step in the KBMS approach, but it means that the application need have no knowledge of whether the prices are listed in a table or calculated routinely, or are dynamically generated objects from the knowledge base. This difference in orientation -- for better or worse -- defines KBMS. In traditional AI terms, KBMS has all the usual good things: backward and forward chaining (goal- and data-directed reasoning), hypothetical reasoning, and object-oriented programming, with inheritance and tactical reasoning (that is, triggers, as opposed to strategic reasoning, which is closer to an expert system with a pre-set agenda mainteained with scheduling and priorities). To users who have experimented with ART or KEE, KBMS looks like what they like about those systems, but in a familiar environment. To traditional dp folks, used to procedural code, KBMS represents a new way of thinking.

By comparison with its competition, KBMS seems to be slightly less portable and robust in a trade-off for better performance (by choosing cross-memory services instead of LU 6.2) and a closer adherence to AI paradigms than ADS and Application Expert, which will look more familiar to traditional MIS folks.

In essence, KBMS is for customers who want the ability to embed expert systems in the application environment, yet to keep the knowledge base logically separate so that it can easily be inspected, maintained, and understood. Rather than a list of rules, it consists of intelligible data definitions (objects), rules, and a control structure that defines how some of the rules (the "strategic" ones) should be executed. Finally, it works in conjunction with AICorp's Intellect, so that much of the information can be expressed in English, both by the user and by the machine. Pricing starts at $90,000.

NAMED PIPES AND ALL THAT

Networking supports the groupware and databases described above. We once said that networking should be like mayonnaise: You know it's there, and its presence enhances things, but it's not something you generally ask for specifically, such as roast beef or tunafish. The proper role of networking is to be supportive and invisible.

But mayonnaise has to be mixed very carefully or it separates into oil and curds; only the mass-produced commercial stuff is a standard, homogeneous product. Likewise, networking won't get smooth until it's a mass-producible product... We wish we didn't have to know about it; such things matter only when they don't work. They don't work, of course, when they're not widespread enough to count on. Here we examine just a few constituent ingredients of the network mayonnaise until, some day, we can buy it off the shelf.

LAN servers

What is the difference between a terminal and a host, on the one hand, and a client and a server, on the other? Relative social standing. A terminal is assumed always to be dumb, whereas a client is a smart machine requesting something that it is capable of handling intelligently. Clients and servers can change roles, depending on who's doing the asking and who's got the server software. The server is the server not because it is so much smarter, but just because it knows how to handle requests from multiple clients.

How does a pc become a server? First, it gets network software, which makes it into a workstation (by many people's definition) able to send messages and request information from other workstations or servers. This component typically consists of Microsoft's Redirector (or IBM's similar Requester), a component that knows how to redirect local requests for services to the appropriate place in the network. Examples of network software include 3Com's 3+, Novell's Netware, and IBM's PC LAN program.

In addition, the server needs server software. That is provided in IBM's PC LAN 1.3 (for DOS-based servers), IBM's LAN Server (separate from Extended Edition, which provides only the workstation part of network support), Microsoft/3Com's LAN Manager (3+ open), and Novell's Netware (server plus client support), among others.

Generally, all the facilities on a workstation become available as network services once the server software is installed, so that a local database becomes a database server, communications facilities become a gateway for remote access, and so forth. Thus will IBM's Extended Edition, initially a way for a single workstation to manage data and talk to remote systems, become a database and communications server with the use of LAN Server.

Network operating systems

Until now, with no standard multi-tasking OS for the PC, Intel-architecture server vendors faced a vacuum and had to build their own non-standard operating systems (except for UNIX, adopted by Banyan). In essence, each vendor built its own kludged-up, limited-function equivalent of OS/2, but now there is a putative standard with OS/2 itself.

3Com has jumped to support OS/2, while Novell has allowed that it will support OS/2 by letting its own Netware OS run as a task under OS/2. Novell contends that OS/2 can't be easily optimized for network management. As a task under OS/2, Nondedicated Netware runs as a single thread, like Sybase, and manages resources more efficiently because it's optimized for network management rather than general functionality. Presumably the greater efficiencies of Netware will offset the overhead of this arrangement; also, you can run OS/2 and its applications, and Netware separately on co-processors or on separate servers on a single network. We don't think Netware's inherent virtues will matter much more in the marketplace than those of Concurrent CP/M did. Most people aside from Novell and its thousand-plus existing ISVs will end up supporting OS/2 only, while Novell will provide support for its large base of current users.

The role of OS/2

One role of OS/2 is simply to be a common operating system for network servers. It doesn't have much in particular to do with networking, but it's important for servers to use it so that the same applications will work regardless of the specific network environment. Aside from being a reasonably agreed-upon standard (contending with UNIX plus a number of proprietary OSes), OS/2 provides a number of useful functions. For example:

- *Multi-tasking*. This is the most significant change from DOS, letting the server support multiple users doing things simultaneously.

- *Interprocess communications*. Interprocess communications let applications communicate; redirectors let them do so across a network.

- *DOS support*. OS/2, for better or worse, handles most of the DOS API. This has burdened OS/2 with extra support facilities, but it vastly expands its utility (and marketability) in the real world.

- *Large memory*. In practice, 640K is ridiculous for a system supporting several users. Larger memory enables the system to perform its various roles and to do the larger processing jobs expected of a server. It also allows for cache memory, vital for good performance.

- *Large file size*. Large file size makes everything more flexible and easier to manage.

- *Protect-mode operation*. This insulates the system from software errors, which can destroy only their own partition of the system.

- *Modularity*. Because the OS consists of modular chunks, it is easy to install and modify extensions without much disruption. Examples are dynamic link libraries and the (forthcoming) installable file system.

Pipes and protocols: Local and remote

One of the difficulties in discussing communications facilities is that they do different things. The OSI reference model, for example, is not really a spec for how to perform a variety of communications functions; it's merely a spec for how to classify those functions. Left unspecified, and thus nonstandard, is how to carry out any particular function. Only the division of labor is clear, but that's a start.

Take, for example, named pipes vs. APPC/LU 6.2 (Advanced Program-to-Program Communications/Logical Unit 6.2). Although they solve different problems in different ways, there is enough overlap that they are competing for support and attention from ISVs and from IBM and Microsoft. Named pipes is the basic Microsoft network solution; APPC/LU 6.2 is IBM's. There is considerable controversy over the relative merits of either approach, and widespread speculation over whether IBM will support named pipes in its next release of OS/2 (which would make it generally available for application developers to rely on). Novell and 3Com have announced that they will support both.

To be elementary, there are two kinds of communications: where the software knows what it will find at the other end, and where it doesn't. Where it knows, things are pretty simple. The limiting case is within a single process, where you're talking to a known thread within your own environment. All you need is a clear channel; you know that what you send through will be properly interpreted by a known thread at the other end. *That* is an "anonymous" pipe -- a channel through which a process can send something internally. When you want to talk to another process or another application, the pipe needs a name to identify it so it can sit anywhere and the sending and receiving applications will be able to find and use it (with help from the OS that supports named pipes). If you want to work across a homogeneous network, you need a transport mechanism under the pipe, but the pipe itself won't look or behave any differently.

Beyond the pipes

But suppose you now want to talk to some other environment. You want a particular service outside of OS/2-Netware. In this case, you can't just send a "local" command down a pipe; you must establish a connection using a specific set of commands for execution by a cross-environment transport that knows how to navigate foreign territory. In this case, the command set goes by the name APPC/PC, and they are interpreted by a logical device called LU 6.2. Translated, those commands can be understood by any other LU 6.2 and translated in turn into the API (application program interface) of the receiving application and environment. (You could run a named pipe over APPC, where APPC/LU 6.2 would provide merely the transport without need for protocol translation because it would address the same environment at the other end. Of course, you'd get the protocol translation, and translation back, whether you needed it or not -- with substantial overhead.)

Thus APPC is a more general approach than the use of named pipes, which assumes a homogeneous environment; on the other hand, it's more demanding in that it requires the application to use a specific set of verbs in writing to the LU 6.2 application program interface, which varies according to the IBM environment you're in. You can write an application that uses APPC verbs and run it within a homogeneous environment or even on a single machine without sacrificing much more than a bit of overhead (both in execution and in the use of several hundred K of memory), and then you could move that application with no changes to run across a heterogeneous environment. (The APPC verbs, however, have mostly to do with the communication itself; the substance of the application is expressed in its own fashion.)

For developers who expect to limit the operation of their applications to OS/2 or Netware, named pipes are a simple but limiting solution. By contrast, APPC is an upfront investment that gives you greater flexibility down

the road (but doesn't make much sense within DOS or on a non-server workstation because of memory requirements). APPC enforces a discipline that comes in handy when you want to extend your reach. Of course, the APPC promise of generality is only as good as the breadth of APPC's spread through the world, which is still limited but potentially extends throughout IBM's Systems Application Architecture. Outside vendors, Apple and Wang among them, are also supporting APPC, as are third-party vendors such as Orion Network Systems, supplier of Apple's APPC implementation among others.

How does LU 6.2 work?

While named pipes offer the application developer the freedom to declare a pipe and then use it, the construct requires a high degree of coordination from both sides. There's nothing in the system that declares the other "end" of the pipe to be ready to receive. By contrast, APPC is like good underware -- supportive and restrictive. It consists of a number of components: First, there's a set of communications verbs, including handshaking routines to ensure that contact has been established. Second, there's LU 6.2), a virtual device with a programming interface (verb set) specified for each environment that supports APPC. (Long-run, these APIs will become more similar, but for now the LU 6.2 API is different in each environment.) Finally, there's the code that creates the LU 6.2, which is responsible for converting the local application's syntax into APPC code, understandable by the receiving LU 6.2 in an arbitrary other environment.

Metaphors

One of the problems of dealing with networks is that we haven't yet settled on appropriate metaphors to describe them to users. The purpose of a metaphor goes beyond putting a cute picture on the screen: It suggests to a user how to use a function and what it does; it suggests to the developer how it might be extended. We have paper, and spreadsheets, and cardfiles for single-user tasks. We even have phone messages and buckslips, but there are few clear metaphors for network services. If networks are truly invisible, of course, we don't need them, but occasionally a user does need to be conscious of a network -- to register his own movements, to inquire about the availability of services, to create a workgroup to receive mail, to establish a connection, etc. How do we show that something is available, but likely to be slow (turn it grey, cover it with a meter?) We may not want to shield the user fully from the import of his actions, especially when they may waste valuable resources.

What are the appropriate metaphors? The word "services" comes to mind, with all that that implies. You can own things -- local peripherals, for example -- or you can rent them from a service. Meters seem appropriate, as do take-a-number systems (where there's limited access). But mostly it seems to make sense to have the hardware itself represented: Perhaps not workstations, per se, but trees of user groups (sort of an org chart, with a "you are here" sign). The same model for data would show what kinds of files are where. For processing services we may need clever icons: Greek for a number-cruncher, a wizard for an expert system (yecch!). For connection services, lines and boxes are as good as we'll get, probably. Or tin cans and string?

More recipes next month....

USER INTERFACES: DON'T MANIPULATE; DELEGATE!

Direct manipulation is the user's feeling that he's working with software intimately, rather than giving it commands. It's the difference between steering a car, and telling it go-left, go-right. With all the homage to user-friendliness over the past few years, it is accepted that users prefer direct manipulation, but how then to account for cruise control?

Wouldn't it be nice to be able to map out the car's path beforehand, and then relax and enjoy the ride? Or better yet, tell it where to go, and have it figure out the route itself, obeying traffic laws and traffic signals with unfailing accuracy? (Of course, we want to retain the ability to stop and make a detour if we see a particularly appealing sight in mid-journey.) We're getting to the point in software where users can learn to read maps and plan routes, if we make it easy.

Half the trick in providing delegation-through-software is abstraction, so the user doesn't have to bother with the details. The other half is giving him power without taking away granularity of control. He doesn't want the software to make all the choices; he wants to be able to specify some choices and then have the software carry them out, intelligently. Imagine giving a babysitter some instructions, like where to call you and the child's bedtime, plus what it's allowed to eat. You still expect the baby sitter to figure out how to put the child to bed, to remember to brush its teeth, and to know what to do in case of fire. Likewise, software should be able to integrate your specific instructions with some general knowledge of its task area, and with some general rules of behavior.

Some examples: *Financial planning languages, where the user specifies the formulas and loads the data automatically, vs. spreadsheets.* Much to many people's surprise, financial planning languages came first. But in the old days they were inflexible and cryptic -- lots of power in the system but little power for the user (which is why most people never used them). Javelin, an excellent product that has failed to find its market, is a fine example of the integration of the two paradigms, as is Lotus's own HAL, with its ability to guess intelligently, select a range and draw a graph when the user says, "Graph this." With better conventions for "naming" data and more flexible ways to categorize data, such as object-oriented data structures, such tools are becoming easier and easier to build. Just wait for Lotus's new set of tools, or ON Technology's new information service.

Forms managers. Forms management is now a hot area. Systems such as Blankity-Blank let you set up a questionnaire and a related data file, then automatically produce a report in yet another format. (This is just a special-purpose form of data entry and report-writing, but the forms packages make it easier.) Automating another part of the task (one painful to do otherwise), SoftView's forthcoming FormSystem can automatically reshape a form to fit changes in the amount of data, column widths or other parameters.

Most CASE tools. What is a CASE tool but a system that helps you specify something abstractly, and then either checks your syntax, logic and consistency (design and analysis tools) or does the lower-level implementation automatically (code and application generators)? Other CASE tools monitor and report on the progress of projects, automatically checking for all the things you're too busy to watch out for. In one sense, CASE tools are

applications that use the delegation-through-software paradigm; in another sense, they are the delegation tools themselves.

A new form of intelligent mail-merge, where the system worries about layout, justification, etc. Tell it what data you want, and then let it format the results correctly according to a sample layout you give it. When we do form letters, we want them centered in the middle of the page, regardless of whether we add a lot or just a little text. Someday, we'd like the system to know how to do a two-page letter, but only as appropriate. Know of any? Right now, we just do the one-pagers and two-pagers in two separate batches.

Desktop publishing. This could solve our mail-merge layout problem, but it's not yet integrated with the mail-merge solution in most reasonably priced systems. The best systems (Ventura Publisher and new releases of Aldus PageMaker) let the user describe general page templates and then automatically lay the text out page by page (vs. the one-by-one page layout approach of the original PageMaker -- and the manual approach it copies).

High-end electronic publishing. As it happens, we saw some more powerful examples of delegation-through-software at the recent Seybold conference on electronic publishing (not desktop publishing -- this is the one where they have newspapers and documentation mass-producers such as Boeing). Style sheets, or collections of formats, which Interleaf calls templates, are a first approximation of this idea; in a rote way, they let the user say: Give me the format I used last time. From the other direction, there are page templates, preconfigured layouts into which a user can drop text (with or without his own modifications), available from Aldus and others.

One level beyond is what DocuPro calls "publishing by example" -- a neat name for what is still a slightly confusing capability. It enables a user to capture the essence -- both formats and the *sequence* of formats -- of any model document, and then reuse them without regard to the length of text. Obviously, this makes more sense in the world of multi-page documents than page templates; it's essentially the same idea, with pagination intelligence thrown in. (We do something of the sort each month by copying our previous issue, deleting most of the text, and saving our headers, footers, copyright notices, etc. If we *had* fancy formats, we'd save those too.)

More impressive yet (and costly, at $1 million and up for a typical installation) is Directory Expert from Expert Technologies, Inc., a Yellow Pages assembler which enables the user to express and execute an arbitrarily rich set of rules directing the placement of objects upon a page. The first rule decrees objects shall fill each page. These "objects" are generally display ads, interspersed with alphabetically sorted telephone listings and, as a last resort, filler ("Let your cursor do the walking" or "Read **Release 1.0**") which doesn't generate revenue. Rules govern such concerns as placing an ad on the same page where its listing appears, putting the big ads below the small ones on the same page, keeping a section together to the extent possible, and starting a new section with listings rather than with an ad. Directory Expert comes with generic rules that the builder-user may alter to taste (also determining which constraints to relax first). It runs on TI Explorers or Suns and is written in LISP; we imagine it will eventually show up on the new TI-Macintosh. Directory Expert can do a ten-pound Yellow Pages in minutes, not hours, and is currently used by three of the largest Yellow Pages publishers. Obviously, the underlying technology is extensible

for catalogues, listings, and other publications where vast amounts of constantly changing data-based text must be assembled and published.

We remember a similar system, nicknamed thINKer, from Composition Systems Inc. (see **Release 1.0**, 21 April 1986), which included multiple page layout among many functions. The company was bought by Crosfield Systems and the project was abandoned before completion. We hope someday to see it or a successor revived. In the meantime, Directory Expert attacks a barely manageable part of the problem effectively.

We have carefully avoided discussing technology in this essay, but it's obvious that most of the systems mentioned here could qualify as expert systems: They have rules for handling a variety of circumstances, and they embody expertise. Some are written in LISP; others in C or COBOL. Any system that lets you give it rules that depend on the data it encounters -- and not just the data itself, but data *about* the data, such as how large it is or how many lines it takes -- may qualify as useful. That's what counts.

INTERFACE ISSUES: TANDY'S DESKMATE -- 'WINDOWS LITE'

Just as it takes a tough man to make a tender chicken, so does it take an especially smart design team to create computers for less-smart people. While certain high-end vendors are spending millions on high-end environments only a few hundreds of thousands will buy, Tandy has spent millions to create a low-end environment targeted at millions of people -- its DeskMate window/menu interface. In a few weeks, the company will be announcing licensing deals with some applications vendors. As for hardware vendors, we sense that's under consideration. As Sun and AT&T know, and as Tandy and IBM might consider, the best way to exploit a technology is generally neither to hoard it nor to share it, but to resell it at a reasonable price.

Like Windows, DeskMate is a combination of device drivers and bit-map graphics and other primitives, coupled with design rules third parties pledge to adhere to. Unlike Windows, it doesn't multi-task other applications and it runs fine in 384K. True, DeskMate may strike some people as "very basic," as Tandy's Ed Juge politely puts it, especially in the face of the Mac and OS/2, but consider how many people bought the Apple II. A lot of people will take very basic at $599 (a Tandy 1000 with DeskMate in ROM) versus elegant at ten times the price.

Release 1.0 is published 12 times a year by EDventure Holdings, 375 Park Ave., New York, NY 10152; (212) 758-3434. It covers the pc, software, CASE, groupware, text management and connectivity markets, and artificial intelligence. Editor & publisher: Esther Dyson; associate publisher: Sylvia Franklin; circulation manager: Hyacinth Frederick; secretary: Anita Fowler; copy chief & consulting editor: William M. Kutik. *Copyright 1988, EDventure Holdings Inc. All rights reserved. No material in this publication may be reproduced without written permission; however, we will gladly arrange for reprints or bulk purchases.* *Subscriptions cost $395 per year, $475 overseas.*

CASE IN CONTEXT

Can there be a market for a product no one can define? Two recent studies of putative CASE users found that there was widespread disagreement between the studied and the studiers on topics so basic as whether or not they were using CASE, and which kinds of tools. Even though Software Magazine gave its respondents a glossary of terms to help them out, says director of market research Bill Gannon, "there was widespread confusion."

That little discrepancy aside, it turns out that CASE has effectively infiltrated the majority of corporations studied (not representative samples, to be sure), although generally only on a couple of projects, or among a few people. In particular, the Schubert Associates study, which conducted sit-down interviews with 100 leading companies in the use of dp, found that users were already seeing tangible benefits -- primarily in accelerated application development time and increased user acceptance. Another major benefit -- reduced maintenance -- hasn't had time to come into play yet.

In general, says Patricia Burke, consulting associate at Schubert, "Everyone knows they need it, even if they're not sure what it is. They're still hoping that if they fool around with it long enough, the magic tool will show up. There's not a lot of brand loyalty at this point; people still see their tools as experimental or pilot efforts."

The underlying factor spurring the adoption of CASE, Burke continues, is a new attitude on the part of non-MIS management. They no longer regard systems development as a mystical process, immune from measurement, cost control, or productivity improvement. They've read enough magazines to know about CASE, and they want some. In a word, they're more willing to meddle. (Some of them even used to be MIS folks themselves -- for example, Federal Express president Jim Barksdale.) By and large, general management loves CASE. The trick is getting programmers to use it; they consider themselves artists, not craftspeople.

Part of the demystification is due to the proliferation of PCs. Burke quotes one manager: "If my secretary can build this for me, why can't MIS build something like it for the company?" Long run, with easier-to-use software, expert systems, rich data structures, and the like, anyone may be able to design systems, letting the machines do the coding. However, Gannon notes, "A lot of these tools are just helping bad designers design bad programs more quickly." However, the ultimate power will still rest with people who can think.

For the numbers, see *Computer-Assisted Software Engineering: Vendor and user issues in an emerging market*, by Schubert Associates (Patricia Burke, (617) 338-0930), and *CASE 1987: Opportunities and competitive analysis in the Computer-Aided Software Engineering market*, by Software Magazine's Sentry Research division (Bill Gannon, (617) 366-2031). Both of these are expensive sponsored studies, but you can get just enough information to whet your curiosity by calling the vendors.

RELEASE 0.5: PSSST! WANNA SEE SOME TABLES??

You've seen them before: Some enterprising soul schleps around town collecting menus from all the best restaurants and (with permission) slaps them into a book. Very useful -- one metric of good advertising is information intensity. Now you can have the same for most computer publications. Most people subscribe to only a few -- PC Week, PC World, Lotus Magazine, etc. We, it seems, get them all -- and they sit on the floor. In either case -- if you miss them by not subscribing or by not having time even to glance at the tables of contents -- PC Contents[tm] performs a useful service with a monthly booklet of reproductions of the contents pages of leading journals.

Publishers like it too, of course. We hasten to add, partly by way of disclaimer, that when we saw this product we called up PCC's Fred Holborn begging to be included. (Now we'll have to make our table of contents less cryptic and more informative. But Juan and Alice will live on inside!)

SOFTEACH

If you got the feeling that the PC Forum is really just a vaporware festival (there, we said it first!), focused on products five years out, then you should try Softeach as an antidote. We attended part of the New York Softeach last month, and it brought us sharply back to reality. There are about a thousand dealers, trying to figure out what will sell, and how to position products against competing products. Spiffs (little extras for dealers), terms and conditions, inventory management and other such mundane issues are all of intense interest; so are T-shirts. For what it's worth, the most crowded sessions were those of Microsoft (Excel), Ashton-Tate (dBASE IV and the new Mac products), and WordPerfect (the whole line). 3Com, which has newly signed with Softsel, wasn't represented, but it should be at later sessions this season. (See calendar.)

TITLES WE'D LIKE TO SEE

LISP hacker dies in nuclear holocaust, from the Expert Systems Journal
OS/2 made easy, volume III
UNIX: The five most important versions, by Vittorio Cassoni
The future of DOS, by Jean-Louis Gassee (3 pages)
Object-oriented hairstyling, by Yves Claude
The joy of interprocess communications
She loves me, she loves me not; a poem in binary
Microsoft and the UNIX question, by the Council on Foreign Relations
Desktop Madness, by the National Council of Luddites
Desktop criticism, by the staff of the Harvard Business Review
Desktop markup, a primer on SGML
Desktop cuisine, by Barry Wine
Desktop sleep, by the release 1.1 team
Desktop sheep, by Gordon Ewe
Desktop Katz, by the former Mrs. Katz
Desktop blues, by Janis Joplin
Desktop travel, a monthly feature in Holiday magazine
How to dress for success in software, by Mitch Kapor
Juan and Alice tell all

RESOURCES & PHONE NUMBERS

Bob Goldman, Larry Harris, **AI Corp.**, (617) 890-8400
K. C. Branscomb, **Aion**, (415) 328-9595 851-3087
Jean-Louis Gassee, Larry Tesler, **Apple Computer**, (408) 973-2590
Ed Esber, **Ashton-Tate**, (213) 538-7354
John Landry, **Cullinet**, (617) 329-7700
Ed Adams, **DB/Access**, (408) 255-2920
Farouk Arjani, Austin Ford, **DocuPro**, (415) 960-1214
Carlos Bhola, Jim Gay, **Expert Technologies**, (412) 621-0818
Jerry Kaplan, **GO Corporation**, (415) 543-3200
Umang Gupta, **Gupta Technologies**, (415) 321-9500
Ray Ozzie, **Iris Associates**, (617) 692-2800
Jim Manzi, Tom Smaldone, Ed Belove, **Lotus Development**, (617) 577-8500
Rob Glaser, Bill Gates, **Microsoft**, (206) 882-8080
Jim Groff, Paul Weinberg, **Network Innovations**, (408) 257-6800
Craig Burton, **Novell**, (801) 379-7950
Mitch Kapor, Peter Miller, **ON Technology**, (617) 225-2545
Pete Tierney, **Oracle**, (415) 598-8157
Paul Rampel, **Orion Network Systems**, (415) 649-4000
Fred Holborn, **PC Contents**, (714) 854-8513
Ann Winblad, **Schubert Associates**, (415) 567-4630
Lynda Schubert Bodman, Pat Burke, **Schubert Associates**, (617) 338-0930
Susan Morgan, **SoftView**, (805) 388-2626
Bob Epstein, **Sybase**, (415) 548-4500
Gordon Eubanks, **Symantec**, (408) 253-9600
Spencer Leyton, **Symantec**, (415) 964-6300
Howard Elias, Ed Juge, **Tandy Corp.**, (817) 390-3549
Bill Krause, **3Com**, (408) 562-6460
Kamran Kheirolomoom, Mike Crosno, **ViewStar**, (415) 841-8565

COMING SOON...

- *Database-based e-mail.*
- *Sun and its planets.*
- *Database server styles: Sybase, Oracle, IBM et al.*
- *Parallel processing.*
- *Nitty-gritty experts.*
- *Airline experts.*
- *And much more...*

RELEASE 1.0 CALENDAR

March 19-20 — **Softeach** - San Francisco. See page 19. Sponsored by Softsel. Contact: Anyone at (800) 325-9189. In Missouri, (314) 827-1724; in Canada, (800) 268-1220.

March 20-23 — **ADAPSO SPRING CONFERENCE** - Palm Desert, CA. Software and services vendors at the oasis. Contact: Sheila Wakefield, (703) 522-5055.

March 22-24 — **AAAI spring symposium** - Palo Alto. Explanation-based learning, parallel architectures, and other topics. Contact: Claudia Mazzetti at AAAI, (415) 328-3123.

March 23 — **Bill Lowe at the Boston Computer Society** - Boston. Come see IBM's new glasnost in action, with Bill Lowe discussing the future of OS/2 and the PS/2. Contact: Janet Cole, (617) 367-8080; or The Recording, (617) 227-0170.

March 23-25 — **CONFERENCE ON OFFICE INFORMATION SYSTEMS** - Palo Alto, CA. Sponsored by IEEE and ACM groups. With Terry Winograd and others. Contact: Robert Allen, (201) 829-5315.

March 27-30 — **Software Publishers Association spring conference** - Berkeley, CA. Contact: Jackie McDonald, (202) 452-1600.

March 28-31 — **World Congress on Computing** - Chicago. Interface Group's response to NCC. Contact: Jane Wemyss, (617) 449-6600.

March 28-31 — **Electronic imaging '88** - Anaheim. Everything you could possibly want to know about image acquisition, processing, storage and retrieval. With speeches by Versatec founder Renn Zaphiropoulos and Rafael Gonzalez, plus sessions by a host of industry experts. Sponsored by MG Expositions Group. Contact: Maureen Gradey or Ed Martin, (800) 223-7126 or (617) 232-3976.

April 7-10 — **13th West Coast Computer Faire** - San Francisco. Contact: Jason Chudnofsky at Interface Group, (617) 449-6600.

April 9-10 — **Softeach** - Chicago. See page 19. Sponsored by Softsel. Contact: Anyone at (800) 325-9189. In Missouri, (314) 827-1724; in Canada, (800) 268-1220.

April 10-13 — **AEA Conference** - Washington, DC. Companies under $75m in annual revenues. Call John Baumeister, (408) 987-4200.

April 10-13 — **Design management west** - San Diego. Call Earl Powell, (617) 236-4165.

April 11-14 — **AIIM show** - Chicago. Information and image management. Sponsored by Association for Information and Image Management. Call Sue Wolk or Betty Garrett, (301) 587-8202.

April 11-15 **IEEE Tenth international conference on software engineering** - Singapore. From an international perspective. Sponsored by IEEE and NCB Singapore. Contact: Tan Chin Nam or Lim Swee Say, (65) 772-0200.

April 19-21 **CEPS/Spring '88** - Chicago. Corporate electronic publishing systems. Sponsored by Cahners and InterConsult. Call Mike Driscoll, (203) 964-0000, or Paula Wertman, (617) 547-0332.

April 20-22 **ICP million-dollar awards** - Atlanta. Sponsored by International Computer Programs, Inc. Call Judy Fary or Stephanie Smith, (800) 426-6179 or (317) 844-7461.

April 23-24 **Softeach** - London. See page 19. Sponsored by Softsel. Contact: Anyone at (800) 325-9189. In Missouri, (314) 827-1724; in Canada, (800) 268-1220.

April 27-29 **Seybold Technology Forum** - Cambridge, MA. "Distributed network computing: A journey into the future." Sponsored by Patricia Seybold's Office Computing Group. Discussions ranging from communications protocols to computer-supported cooperative work. Call Catherine Cooper, (617) 742-5200.

May 3-5 **Expert systems and the leading edge in production planning and control** - Charleston, SC. Nitty-gritty experts par excellence, with talks by Tom Kehler and others. Contact: Libby Shropshier, (803) 777-5766.

May 3-6 **CASExpo** - Dallas. Managed by Arthur Young & Co. Contact: Ken Burroughs, (703) 845-1657.

May 4-6 **AI Long Beach** - Long Beach, CA. The most commercial AI show around, for better or worse. Sponsored by Tower Conference Mgt. Call Caroline Jackson, (312) 469-3373.

May 9-12 **Comdex Spring** - Atlanta. Peaches and PCs. Contact: Jane Wemyss at the Interface Group, (617) 449-6600.

May 10-12 **Computer services and consultants executive conference** - Palm Springs. IBM mingles with its third-party support cadres. By invitation only. Contact your own contact or call Elsie Chapman, (914) 749-3389.

May 12-13 **Computer Law Institute** - Los Angeles. The ninth annual, sponsored by the Universtiy of Southern California. Panels on intellectual property in the U.S. and abroad, new applications, the IBM/Fujitsu case, liquidity for high-tech companies. Contact: Jerry Wiley, (213) 743-2582.

May 15-18 **Software Maintenance Association annual meeting and conference** - Chicago. Feeling good about software maintenance.... With Robert Glass and Gerald Weinberg. Contact: Judith Marx Golub, (415) 852-1640.

May 15-19	**Human factors in computing systems** - Washington, DC. Sponsored by ACM groups and the Human Factors Society. Contact: Sylvia Sheppard, (301) 369-2422, Scott Robertson, (201) 932-2911, or ACM (212) 869-7440.
May 15-20	**International Communications Association 41st annual conference** - Anaheim. Sponsored by the International Communications Association, with speeches by Peter Keen and Bernard Flanagan, vp marketing of The Wall Street Journal. Contact: Lee FLowers, (214) 233-3889.
June 1-3	**CASE Benchmarks** - Dallas. Rather than just present companies and tools, moderator Vaughan Merlyn controls the proceedings and *compares* the various tools on a common scale. Sponsored by Digital Consulting. Call Scott Dorman, (617) 470-3870.
June 6-8	**Artificial intelligence in electronic publishing** - San Jose. Sponsored by the Graphic Communications Association. Applying AI to design, content, process, etc. Contact: Marion Elledge, (703) 841-8160.
June 13-15	**Conference on R&D in information retrieval** - Grenoble, France. Co-chaired by Gerard Salton; sponsored by ACM. Contact: Gerard Salton, (607) 255-4117.
June 19-22	**Congress VI** - Paris. The world computing services industry gets together. Sponsored by national trade organizations, including our own Adapso. Contact: Phyllis Cockerham, (703) 522-5055, or Diana Kirby, London, (441) 405-2171.
June 21-23	**PC Expo** - New York City. End-users en masse. Contact Steve Gross, (201) 569-8542 or (800) 922-0324.
July 12-15	**CASE '88** - Cambridge, MA. Second international workshop on computer-aided software engineering. More academics and less hype than most CASE conferences, for better or worse. Sponsored by several academic institutions. Call Pamela Meyer, Index Technology (organizers), (617) 494-8200, x454.
July 24-27	**IEEE conference on neural nets** - San Diego. The second, because the first was so successful. Contact: Richard Rea (exhibits), (619) 222-7477, Sue Varga, (619) 281-8991, or Nomi Feldman (papers), (619) 453-6222.
August 1-5	**SIGGRAPH** - Atlanta. Sponsored by IEEE, ACM and SIGGRAPH. Contact: Adele Newton, (519) 888-4534.
August 8-12	**IEEE conference on applications of artificial intelligence in engineering** - Stanford, CA. With Raj Reddy and Rick Hayes-Roth, among others. Sponsored by IEEE. Contact: R. Adey, (617) 667-7582.
August 22-26	**AAAI-88** - St. Paul, MN. The seventh annual. Sponsored by the American Association for Artificial Intelligence. Contact: Claudia Mazzetti, (415) 328-3123.

SUBSCRIPTION FORM

Please enter my subscription to **Release 1.0** at the rate of $395 per year in the U.S. and Canada. Overseas subscriptions are $475, airmail postage included. Payment must be enclosed. Multiple-copy rates on request. Satisfaction guaranteed or your money back.

Name _____

Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Telephone _____

How did you hear about Release 1.0? _____

_____

WCC8 88-3

Please fill in the information above and send with your check payable to:

EDventure Holdings Inc.
375 Park Avenue, Suite 2503
New York, NY 10152

If you have any questions, please call us at (212) 758-3434.

Sylvia Franklin
Associate Publisher